み

	Application No.	Applicant(s)
Notice of Allowability	09/858,260	SHAH ET AL.
	Examiner	**Art Unit**	
	Kenny Lin	2154	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *2/22/2005*.

2. ☒ The allowed claim(s) is/are *1-96 now renumberd as 1-96*.

3. ☒ The drawings filed on *15 May 2001* are accepted by the Examiner.

4. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a) ☐ All   b) ☐ Some*   c) ☐ None   of the:

      1. ☐ Certified copies of the priority documents have been received.

      2. ☐ Certified copies of the priority documents have been received in Application No. _____ .

      3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

    * Certified copies not received: _____ .

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application. **THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.**

5. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

6. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.

    (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached

      1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____ .

    (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of

    Paper No./Mail Date _____ .

    Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

7. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☒ Notice of References Cited (PTO-892)

2. ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3. ☒ Information Disclosure Statements (PTO-1449 or PTO/SB/08), Paper No./Mail Date <u>all</u>

4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material

5. ☐ Notice of Informal Patent Application (PTO-152)

6. ☒ Interview Summary (PTO-413), Paper No./Mail Date *4/26/2005* .

7. ☒ Examiner's Amendment/Comment

8. ☒ Examiner's Statement of Reasons for Allowance

9. ☐ Other _____ .

N. El Hady

DETAILED ACTION

1.   Claims 1-96 are presented for examination.

2.   The IDS has been considered by the examiner.

3.   The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Hughes, US 6,854,009.

Brewer et al, US 2002/0078170.

Delo et al, US 6,418,554.

Lucovsky et al, US 6,836,794.

Tzelnic et al, US 5,948,062.

Deo et al, US 6,226,665.

Bonham et al, US 6,157,959.

Cox et al, US 6,510,466.

Keeley, US 6,138,271.

EXAMINER'S AMENDMENT

4.   An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

5.   Authorization for this examiner's amendment was given in a telephone interview with

John King on April 28, 2005.

6.   The application has been amended as follows:

Claims 1, 3 8, 13, 15, 20, 25, 27, 32, 37, 39, 44, 49, 51, 56, 61, 63, 68, 73, 75, 80, 85, 87 and 92

are amended.

1.   (Currently Amended) A process for client-side retrieval, storage, and execution
of application programs and other related data streamed from a server across a computer network
to a client system in a computer environment, comprising the steps of:

        providing a streaming file system on said client;

        wherein said streaming file system appears to install parts of an application
program on said client sufficient for execution to contain an application program;

        while executing the application program, wherein said streaming file system
receives requests from local processes of said client for application program code
[[or]]and data that are part of the application program not already installed when said
application program code and data are needed;

        loading a portion of said application program code orand data into a persistent
cache on said client wherein said portion of said application program code [[or]]and data
stored in said persistent cache is available for reuse after termination of execution of said
application program; and

        wherein said streaming file system satisfies said requests for said application
program code [[or]]and data by retrieving [[it]]said requested application program code
and data from said persistent cache or by retrieving [[it]]said requested application
program code and data from said server without interrupting the execution of the
application program.

3.   (Currently Amended) The process of claim 1, wherein said client initiates the prefetching of application program code and data from said server; and wherein said client inspects program code [[or]]and data file requests and consults the contents of said persistent cache as well as historic information about application program fetching patterns and uses [[this]]said historic information to request additional blocks of application program code and data from said server that said client expects will be needed soon.

8.   (Currently Amended) The process of claim 1, further comprising the step of:
        maintaining checksums of application code and data in said persistent cache;
        wherein when a block of code and/or data is requested by a local process said streaming file system computes the checksum of the data block before it is returned to the local process; and
        wherein if a computed checksum does not match the checksum stored in said persistent cache the cache entry is invalidated and a fresh copy of the page is retrieved from said server.

13.   (Currently Amended) A process for client-side retrieval, storage, and execution of application programs and other related data streamed from a server across a computer network to a client system in a computer environment, comprising the steps of:
        providing a kernel-mode streaming file system driver on said client;
        wherein said kernel-mode streaming file system appears to install parts of an application program on said client sufficient for execution;
        providing a user-mode client on said client;
        while executing the application program,wherein said streaming file system receives requests from local processes of said client for application program code [[or]]and data that are part of [[an]]said application program not already installed when said application program code and data are needed;

loading a portion of said application program code [[or]]and data into a persistent

cache on said client wherein said portion of said application program code [[or]]and data

stored in said persistent cache is available for reuse;

wherein requests made to said streaming file system are directed to said user-

mode client or retrieved from said persistent cache; and

wherein said user-mode client handles the application program code and data

streams from said server and sends the results back to said streaming file system driver

without interrupting the execution of the application program.

15.   (Currently Amended) The process of claim 13, wherein said client initiates the

prefetching of application program code and data from said server; and wherein said client

inspects program code [[or]]and data file requests and consults the contents of said persistent

cache as well as historic information about application program fetching patterns and uses

[[this]]said historic information to request additional blocks of application program code and

data from said server that said client expects will be needed soon.

20.   (Currently Amended) The process of claim 13, further comprising the step of:

maintaining checksums of application code and data in said persistent [[30]]

cache;

wherein when a block of code and/or data is requested by a local process said

streaming file system computes the checksum of the data block before it is returned to the

local process; and

wherein if a computed checksum does not match the checksum stored in said

persistent cache the cache entry is invalidated and a fresh copy of the page is retrieved

from said server.

25.   (Previously Presented) A process for client-side retrieval, storage, and execution

of application programs and other related data streamed from a server across a computer network

to a client system in a computer environment, comprising the steps of:

providing a streaming block driver on said client;

wherein said block driver provides the abstraction of a physical disk to a native file system already installed on the client operating system and appears to install parts of an application program on said client sufficient for execution;

loading a portion of said application program code and data data associated with said application program into a persistent cache on said client wherein said application program code and data stored in said persistent cache is available for reuse;

wherein said block driver receives requests for application program code and data for physical block reads and writes from local processes of said client which [[it]]said block driver satisfies out of said persistent cache on a standard file system that is backed by a physical disk drive without interrupting the execution of the application program; and

wherein said requests that cannot be satisfied by said persistent cache are sent to said server.

27.   (Currently Amended) The process of claim 25, wherein said client initiates the prefetching of application program code and data from said server; and wherein said client inspects program code [[or]]and data file requests and consults the contents of said persistent cache as well as historic information about application program fetching patterns and uses [[this]]said historic information to request additional blocks of application program code and data from said server that said client expects will be needed soon.

32.   (Currently Amended) The process of claim 25, further comprising the step of:

maintaining checksums of application code and data in said persistent cache;

wherein when a block of code and/or data is requested by a local process said streaming file system computes the checksum of the data block before it is returned to the local process; and

wherein if a computed checksum does not match the checksum stored n said persistent cache the cache entry is invalidated and a fresh copy of the page is retrieved from said server.

37.   (Currently Amended) A process for client-side retrieval, storage, and execution of application programs and other related data streamed from a server across a computer network to a client system in a computer environment, comprising the steps of:

        providing a disk driver on said client;

        providing a user-mode client on said client;

        wherein said user-mode client appears to install parts of an application program on said client sufficient for execution;

        while executing the application program, wherein said disk driver sends all file requests that [[it]]said disk driver receives to said user-mode client; requesting for application program code and data that are part of said application program not already installed when said application program code and data are needed;

        loading a portion of said application program code and data data associated with an application program into a persistent cache on said client wherein said application program code and data stored in said persistent cache is available for reuse; and

        wherein said user-mode client attempts to satisfy said file requests from said persistent cache or from said server without interrupting the execution of the application program.

39.   (Currently Amended) The process of claim 37, wherein said client initiates the prefetching of application program code and data from said server; and wherein said client inspects program code [[or]]and data file requests and consults the contents of said persistent cache as well as historic information about application program fetching patterns and uses [[this]]said historic information to request additional blocks of application program code and data from said server that said client expects will be needed soon.

44.   (Currently Amended) The process of claim 37, further comprising the step of:

        maintaining checksums of application code and data in said persistent cache;

wherein when a block of code and/or data is requested by a local process said streaming file system computes the checksum of the data block before it is returned to the local process; and

wherein if a computed checksum does not match the checksum stored n said persistent cache the cache entry is invalidated and a fresh copy of the page is retrieved from said server.

49.   (Currently Amended)   An apparatus for client-side retrieval, storage, and execution of application programs and other related data streamed from a server across a computer network to a client system in a computer environment, comprising:

a streaming file system on said client;

wherein said streaming file system appears to install parts of an application program on said client sufficient for execution to contain the installed application program;

while executing the application program, wherein said streaming file system receives requests from local processes of said client for application program code or data that are part of an application program not already installed when said application program code and data are needed;

a persistent cache on said client wherein the persistent cache stores a portion of said application program code [[or]]and data, and wherein said portion of said application program code [[or]]and data is available for reuse after termination of execution of said application program; and

wherein said streaming file system satisfies said requests for said application program code [[or]]and data by retrieving [[it]]said requested application program code and data from said persistent cache stored in a native file system or by retrieving [[it]]said requested application program code and data from said server without interrupting the execution of the application program.

51.   (Currently Amended) The apparatus of claim 49, wherein said client initiates the prefetching of application program code and data from said server; and wherein said client inspects program code [[or]]and data file requests and consults the contents of said persistent cache as well as historic information about application program fetching patterns and uses [[this]]said historic information to request additional blocks of application program code and data from said server that said client expects will be needed soon.

56.   (Currently Amended) The apparatus of claim 49, further comprising:

a module for maintaining checksums of application code and data in said persistent cache;

wherein when a block of code and/or data is requested by a local process said streaming file system computes the checksum of the data block before it is returned to the local process; and

wherein if a computed checksum does not match the checksum stored n said persistent cache the cache entry is invalidated and a fresh copy of the page is retrieved from said server.

61.   (Currently Amended) An apparatus for client-side retrieval, storage, and execution of application programs and other related data streamed from a server across a computer network to a client system in a computer environment, comprising:

a kernel-mode streaming file system driver on said client;

wherein said kernel-mode streaming file system appears to install parts of an application program on said client sufficient for execution;

a user-mode client on said client;

while executing the application program,wherein said streaming file system receives requests from local processes of said client for application program code [[or]]and data that are part of the application program not already installed when said application program code and data are needed;

a persistent cache on said client wherein said persistent cache has a portion of said application program code [[or]]and data loaded therein and wherein said portion of said application program code [[or]]and data is available for reuse after termination of execution of said application program;

wherein requests made to said streaming file system are directed to said user-mode client or retrieved from said persistent cache; and

wherein said user-mode client handles the application program code and data streams from said server and sends the results back to said streaming file system driver without interrupting the execution of the application program.

63.   (Currently Amended) The apparatus of claim 61, wherein said client initiates the prefetching of application program code and data from said server; and wherein said client inspects program code and[[or]] data file requests and consults the contents of said persistent cache as well as historic information about application program fetching patterns and uses [[this]]said historic information to request additional blocks of application program code and data from said server that said client expects will be needed soon.

68.   (Currently Amended) The apparatus of claim 61, further comprising: a module for maintaining checksums of application code and data in said persistent cache;

wherein when a block of code and/or data is requested by a local process said streaming file system computes the checksum of the data block before it is returned to the local process; and

wherein if a computed checksum does not match the checksum stored n said persistent cache the cache entry is invalidated and a fresh copy of the page is retrieved from said server.

73.   (Currently Amended)   An apparatus for client-side retrieval, storage, and execution of application programs and other related data streamed from a server across a computer network to a client system in a computer environment, comprising:

a streaming block driver on said client;

wherein said block driver provides the abstraction of a physical disk to a native file system already installed on the client operating system and appears to install parts of an application program on said client sufficient for execution;

a persistent cache on said client wherein said persistent cache has a portion of said application program code and data data associated with an application program stored therein and wherein said application program code and data stored in said persistent cache is available for reuse after termination of execution of said application program;

wherein said block driver receives requests for said application program code and data for physical block reads and writes from local processes of said client which [[it]]said block driver satisfies out of said persistent cache on a standard file system that is backed by a physical disk drive without interrupting the execution of the application program; and

wherein said requests that cannot be satisfied by said persistent cache are sent to said server.

75.   (Currently Amended) The apparatus of claim 73, wherein said client initiates the prefetching of application program code and data from said server; and wherein said client inspects program code [[or]]and data file requests and consults the contents of said persistent cache as well as historic information about application program fetching patterns and uses this information to request additional blocks of application program code and data from said server that said client expects will be needed soon.

80.   (Currently Amended) The apparatus of claim 73, further comprising:

a module for maintaining checksums of application code and data in said persistent cache;

wherein when a block of code and/or data is requested by a local process said streaming file system computes the checksum of the data block before it is returned to the local process; and

wherein if a computed checksum does not match the checksum stored n said persistent cache the cache entry is invalidated and a fresh copy of the page is retrieved from said server.

85.   (Currently Amended)   An apparatus for client-side retrieval, storage, and execution of application programs and other related data streamed from a server across a computer network to a client system in a computer environment, comprising:

a disk driver on said client;

a user-mode client on said client;

wherein said user-mode client appears to install parts of an application program on said client sufficient for execution;

while executing the application program,wherein said disk driver sends all file requests that [[ft]]said disk driver receives to said user-mode client requesting for application program code and data that are part of said application program not already installed when said application program code and data are needed;

a persistent cache on said client wherein said persistent cache has a portion of said application program code and data data associated with an application program stored therein and wherein said application program code and data stored in said persistent cache is available for reuse after termination of said application program; and

wherein said user-mode client attempts to satisfy said file requests from said persistent cache or from said server without interrupting the execution of the application program.

87.   (Currently Amended) The apparatus of claim 85, wherein said client initiates the prefetching of application program code and data from said server; and wherein said client inspects program code [[or]]and data file requests and consults the contents of said persistent cache as well as historic information about application program fetching patterns and uses

[[this]]said historic information to request additional blocks of application program code and
data from said server that said client expects will be needed soon.

92.   (Currently Amended) The apparatus of claim 85, further comprising:

a module for maintaining checksums of application code and data in said
persistent cache;

wherein when a block of code and/or data is requested by a local process said
streaming file system computes the checksum of the data block before it is returned to the
local process; and

wherein if a computed checksum does not match the checksum stored in said
persistent cache the cache entry is invalidated and a fresh copy of the page is retrieved
from said server.

Allowable Subject Matter

7.   Claims 1-96 are allowed.

8.   The following is an examiner's statement of reasons for allowance: None of the prior art

of record teaches or fairly suggests all of the claimed limitations, especially to install part of the

application program sufficient for execution, while executing the application program, loading

portions of the application program code and data from persistent cache or server when they are

needed without interrupting the execution of the application program.

Any comments considered necessary by applicant must be submitted no later than the

payment of the issue fee and, to avoid processing delays, should preferably accompany the issue

fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for

Allowance."

9.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Kenny Lin whose telephone number is (571) 272-3968. The

examiner can normally be reached on 8 AM to 5 PM Tue.-Fri. and every other Monday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, John Follansbee can be reached on (571) 272-3964. The fax phone number for the

organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

ksl
April 28, 2005